



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/997,990	11/30/2001	Jeremy Alan Arnold	IBM / 193	4258
7590	05/04/2005		EXAMINER	
Scott A. Stinebruner Wood, Herron & Evans, L.L.P. 2700 Carew Tower 441 Vine St. Cincinnati, OH 45202-2917			PHAM, CHRYSTINE	
			ART UNIT	PAPER NUMBER
			2192	
DATE MAILED: 05/04/2005				

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/997,990	ARNOLD ET AL.	
	Examiner	Art Unit	
	Chrystine Pham	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 03 December 2004.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-6,8-27 and 29-40 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-6,8-27 and 29-40 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>10/08/2004</u> . | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the Amendment filed on December 3rd 2004. Applicants have canceled claims 7, and 28. Claims 1, 8, 14, 23, 29, 34, 38, 40 have been amended. Claims 1-6, 8-27, 29-40 are presented for examination.

Response to Amendment

2. In view of the amendment made to the Specification to include the serial number for the related patent application, objection to the Specification is hereby withdrawn.

Response to Arguments

3. The Applicants essentially contend that West (US 5740440) does not teach setting the "creation breakpoint", which is now presented in amended claims 1, 23, and 38. It is respectfully submitted that this "creation breakpoint", as defined by the Applicants in the Reply dated December 3rd 2004, is considered by the examiner to be just a "conceptual breakpoint". That is to say, the "creation breakpoint" does not constitute a conventional breakpoint as defined and acknowledged in the art since a conventional breakpoint is a set of debugging instructions that can be individually inserted into the program code to transfer execution control of the program code to the debugger. The "creation breakpoint", as defined by the Applicants, is merely a naming convention for a collection "entity" which contains a plurality of breakpoints to be collectively set (as a group) in constructor methods of a class. Thus, the "creation breakpoint" will be interpreted hereinafter, as collectively setting a plurality of breakpoints in a plurality of constructor methods. Applicants' arguments with respect to claims 1, 23, and 38 have been considered but are moot in view of the new ground(s) of rejection.
4. The Applicants further stated in the Reply that Applicant Admitted Prior Art (AAPA), upon which the Examiner relied for the rejection of claim 14, does not disclose associating conditional breakpoints (e.g., breakpoints with hit counters) with breakpoints which are set on constructors as

Art Unit: 2192

disclosed by West. The Applicants had seemed to suggest that setting conditional breakpoints on constructor methods are, somehow, impossible. It is respectfully submitted that setting conditional breakpoints in class methods is a well-known practice in the art of object-oriented software debugging (see for example, *Carmichael et al.*, US 6240545 B1 Abstract for setting conditional breakpoint on all methods of a class). It is further submitted that, constructor methods are inherently, class methods. Thus, it is inherent that breakpoints can be set in constructor methods, be they conditional or unconditional.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-6, 8, 12, 13, 23-29, 33, 38, and 39 are rejected under 35 U.S.C. 102(b) as being anticipated by Lenkov et al. (*Lenkov et al.*, US 5560009).

Claim 1

Lenkov et al. teach a computer-implemented method (i.e., program product) (see at least *debugger 320 FIG.3 & associated text*) of debugging an object-oriented computer program (see at least *source code 102 FIG.3 & associated text*), the method comprising:

- o in response to user input (see at least *user 1118, terminal 218, main debugger 1112 FIG.11 & associated text*), setting a creation breakpoint for a class defined in the object-oriented computer program (see at least *class breakpoints, overloaded functions col.29:15-67*), wherein setting the creation breakpoint includes identifying a plurality of creators (i.e., constructors) for the class (see at least *debug information*

316 FIG.3 & associated text; *all member functions* col.29:15-67; *descriptor table, program 102, C++, functions, overloaded, constructor* col.27:5-12; *overloaded functions* col.29:15-67) and setting (i.e., inserting) a plurality of breakpoints (i.e., debugging program code) on the identified creators, that is to say, tracking object creations resulting from multiple creators (see at least *breakpoint, all member functions* col.29:15-67; *overloaded functions, group, breakpoint* col.29:15-67), wherein each of the plurality of breakpoints is associated with the creation breakpoint (see at least *overloaded functions, group, breakpoint* col.29:15-67); and

- o halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints (see at least *class breakpoint, suspends execution, member functions* col.29:15-67).

Claim 2

The rejection of base claim 1 is incorporated. *Lenkov et al.* teach wherein identifying the plurality of creators includes identifying every creator for the class (see at least *debug information* 316 FIG.3 & associated text; *all member functions* col.29:37-67).

Claim 3

The rejection of base claim 1 is incorporated. *Lenkov et al.* further teach after identifying the plurality of creators, displaying a list of the identified creators (see at least *all overloaded functions* col.29:55-67) and receiving user input to select a subset of identified creators, wherein the plurality of breakpoints are set on only the subset of the identified creators (see at least *any set of overloaded functions, breakpoint, all overloaded functions, user 1118* col.29:55-67).

Claim 4

Art Unit: 2192

The rejection of base claim 1 is incorporated. *Lenkov et al.* further teach wherein the plurality of breakpoints are collectively set on all of the identified creators in response to the user input (see at least *overloaded functions, group col.29:37-67*).

Claim 5

The rejection of base claim 1 is incorporated. *Lenkov et al.* further teach wherein setting the plurality of breakpoints includes setting each breakpoint from plurality of breakpoints on a statement in one of the identified creators (see at least *source lines, source code 102, functions col.27:5-25*).

Claim 6

The rejection of base claim 5 is incorporated. Claim recites limitations, which have been addressed in claim 1, therefore, is rejected for the same reasons as cited in claim 1.

Claim 8

The rejection of base claim 1 is incorporated. *Lenkov et al.* further teach in response to the user input to set the creation breakpoint, adding an entry for the creation breakpoint in a breakpoint data structure (see at least *saving, breakpoint col.29:15-25*), wherein setting the plurality of breakpoints includes storing breakpoint information for each breakpoint in the breakpoint data structure (see at least *saving, breakpoint col.29:15-25*), wherein the breakpoint information for each breakpoint is associated with the entry in the breakpoint data structure for the creation breakpoint (see at least *overloaded functions, group, breakpoint col.29:15-67*).

Claim 12

The rejection of base claim 1 is incorporated. *Lenkov et al.* further teach wherein each creator comprises a constructor method defined in the class (see at least *functions, overloaded, constructor col.27:5-15*).

Claim 13

The rejection of base claim 1 is incorporated. *Lenkov et al.* further teach collectively removing the plurality of breakpoints in response to user input (see at least *deletion, breakpoint* col.29:15-67; *debugger 320, operations, overloaded functions, group* col.29:55-60).

Claim 23

Lenkov et al. teach an apparatus (see at least FIG.2 & associated text), comprising:

- a memory (see at least *RAM 210, Data Storage 222* FIG.2 & associated text) within which resides at least a portion of an object-oriented computer program (see at least *source code 102* FIG.3 & associated text); and
- program code configured to debug the object-oriented computer program by (see at least *debugger 320* Fig.3 & associated text), in response to user input (see at least *user 1118, terminal 218, main debugger 1112* FIG.11 & associated text), setting a creation breakpoint for a class defined in the object-oriented computer program (see at least *class breakpoints, overloaded functions* col.29:15-67), wherein setting the creation breakpoint includes identifying a plurality of creators (i.e., constructors) for the class (see at least *debug information 316* FIG.3 & associated text; *all member functions* col.29:15-67; *descriptor table, program 102, C++, functions, overloaded, constructor* col.27:5-12; *overloaded functions* col.29:15-67) and setting (i.e., inserting) a plurality of breakpoints (i.e., debugging program code) on the identified creators, that is to say, tracking object creations resulting from multiple creators (see at least *breakpoint, all member functions* col.29:15-67; *overloaded functions, group, breakpoint* col.29:15-67), and by halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints (see at least *class breakpoint, suspends execution, member functions* col.29:15-67), wherein each of the plurality of breakpoints is associated with the creation breakpoint (see at least *overloaded functions, group, breakpoint* col.29:15-67).

Art Unit: 2192

Claims 24-27, 29, and 33

Claims recite limitations which have been addressed in claims 2-5, 8, and 13, therefore, are rejected for the same reasons as cited in claims 2-5, 8, and 13.

Claim 38

Lenkov et al. teach a program product (see at least *debugger 320 FIG.3 & associated text*), comprising:

- program code configured to debug an object -oriented computer program (see at least *source code 102 FIG.3 & associated text*) by, in response to user input (see at least *user 1118, terminal 218, main debugger 1112 FIG.11 & associated text*), setting a creation breakpoint for a class defined in the object-oriented computer program (see at least *class breakpoints, overloaded functions col.29:15-67*), wherein setting the creation breakpoint includes identifying a plurality of creators (i.e., constructors) for the class (see at least *debug information 316 FIG.3 & associated text; all member functions col.29:15-67; descriptor table, program 102, C++, functions, overloaded, constructor col.27:5-12; overloaded functions col.29:15-67*) and setting (i.e., inserting) a plurality of breakpoints (i.e., debugging program code) on the identified creators, that is to say, tracking object creations resulting from multiple creators (see at least *breakpoint, all member functions col.29:15-67; overloaded functions, group, breakpoint col.29:15-67*), and by halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints (see at least *class breakpoint, suspends execution, member functions col.29:15-67*), wherein each of the plurality of breakpoints is associated with the creation breakpoint (see at least *overloaded functions, group, breakpoint col.29:15-67*); and
- a signal bearing medium bearing the program code (see at least *100, 122 FIG.1 & associated text; 228, 210, 222 FIG.2 & associated text*).

Claim 39

The rejection of base claim 38 is incorporated. *Lenkov et al.* further teach wherein the signal bearing medium includes at least one of a transmission medium and a recordable medium (see at least 204, *RAM 210 FIG.2 & associated text*).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 9-11, 14-15, 17-22, 30-32, 34-35, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Lenkov et al.* in view of *Phillips et al.* (*Phillips et al.*, US 5321828).

Claim 9

The rejection of base claim 1 is incorporated. *Lenkov et al.* do not expressly disclose tracking a total number of hits to the plurality of breakpoints. However, *Phillips et al.* teach a method of debugging a computer program comprising tracking a number of hits to the plurality of breakpoints (see at least Abstract; *software breakpoints, program, function* col.26:40-col.27:25; *disabling breakpoints, break conditions, ignore* col.28:1-col.29:25). *Lenkov et al.* and *Phillips et al.* are analogous art because they are both directed to debugging computer program. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Phillips et al.* into that of *Lenkov et al.* for the inclusion of tracking the number of hits to the breakpoints. And the motivation for doing so would have been to facilitate the monitoring of data (e.g., program variables, conditions, states) at specific execution points (i.e., each time a breakpoint is hit or encountered) and storing these information for later retrieval, thus enabling the user to specify a special point of interest (e.g., when a specific breakpoint has been encountered n number of times, that is to say, when the instruction, or

function in which the breakpoint was set has been executed or called n number of times) at which he would like to analyze program conditions (see *Phillips et al.* col.26:38-65; col.27:40-col.28:40).

Claim 10

The rejection of base claim 9 is incorporated. *Phillips et al.* further teach halting execution of the computer program during debugging in response to hitting any of the plurality of breakpoints includes: determining whether the total number of hits meets a condition in response to hitting any of the plurality of breakpoints (see at least *disabling breakpoints, ignore, n number of times* col.28:1-col.29:15) and halting execution of the object-oriented computer program if the total number of hits meets the condition (i.e., threshold) (see at least *target system 14 program, breakpoint, interrupt tracing, hit, <count> number of times* col.28:65-col.29:20). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Phillips et al.* into that of *Lenkov et al.* for the inclusion of determining the number of hits meeting a condition and halting execution of the program when the number meets the condition. And the motivation for doing so would have been the same, as has been cited in claim 9.

Claim 14

Lenkov et al. teach a computer-implemented method (program product) of debugging an object-oriented computer program (see claim 1), the method comprising tracking object creations (resulting from multiple creators) of a class defined in the object-oriented computer program (see claim 1). *Lenkov et al.* do not expressly disclose tracking a number of said object creations and halting execution of the object-oriented computer program in response to the number of object creations meeting a condition. However, *Phillips et al.* disclose tracking a number of hits for a breakpoint (see claim 9) and halting execution of the computer program in response to the number of hits meeting a condition (see claim 10). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Phillips et al.* into that of *Lenkov et al.* for the inclusion of tracking a number of object

creations and halting execution in response to the number meeting a condition. And the motivation for doing so would have been the same as has been cited in claim 9.

Claims 11, 15

Claims recite limitations which have been addressed in claim 10, therefore, are rejected for the same reasons as cited in claim 10.

Claims 17-22

Claims recite limitations which have been addressed in claims 1-4, and 12, therefore, are rejected for the same reasons as cited in claims 1-4, and 12.

Claims 30-32, 34-35, and 40

Claims recite limitations which have been addressed in claims 1, 5, 9-11, 14, 23, and 38, therefore, are rejected for the same reasons as cited in claims 1, 5, 9-10, 14, 23, and 38.

9. Claims 16, 36, and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Lenkov et al.* in view of *Phillips et al.* further in view of *Pardo et al.* (*Pardo et al.*, US 5754839).

Claim 16

The rejection of base claim 14 is incorporated. *Lenkov et al.* and *Phillips et al.* do not expressly disclose incrementing a counter in response to hitting any of a plurality of breakpoints. However, *Pardo et al.* disclose incrementing a counter (see at least *counter module 40, counter 41, counter 42 FIG.2 & associated text*) in response to hitting a watchpoint (see at least *breakpoint col.5:37-50; breakpoints, counters 41, 42, watchpoints col.6:15-60*). *Lenkov et al.*, *Phillips et al.* and *Pardo et al.* are analogous art because they are directed to debugging computer program using breakpoints. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Pardo et al.* into that of *Lenkov et al.* and *Phillips et al.* for the inclusion of incrementing a

counter in response to hitting a breakpoint. And the motivation for doing so would have been to improve the debugging program and the information generated thereby. That is to say, keeping a breakpoint counter and incrementing the counter in response to hitting the breakpoint enables the instructions where breakpoints are set to be speculatively executed and their results to be flushed in case of an interrupt without generating false breakpoints (see *Pardo et al.* col.1:15-60; col.2:25-62).

Claims 36-37

Claims recite limitations, which have been addressed in claims 1, and 16, therefore, are rejected for the same reasons as cited in claims 1 and 16.

Conclusion

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 571-272-3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CP
April 22, 2005



TUAN DAM
SUPERVISORY PATENT EXAMINER